Oral History of Earnest E. (Lee) Keet

Interviewed by:
Luanne Johnson

Recorded: April 9, 1986
Houston, Texas

CHM Reference number: X5558.2010

Table of Contents

Ernest E. (Lee) Keet

Conducted by Luanne Johnson

Abstract: Ernest E. (Lee) Keet describes the founding of Turnkey Systems in April 1967 and how their business strategy evolved from professional services to software products over the next several years. He describes the release of their product Task/Master, the first commercially available telecommunications monitor to be launched on a worldwide basis, and how the challenges of meeting customer expectations drove the continuous improvement of the product. He reveals how IBM's policy to require customers to use CICS to support products such as DL1 killed the market for independent vendors of telecommunications monitors and how Turnkey Systems made a decision to take Task/Master off the market as a result. He describes the sale of Turnkey Systems to National CSS which was shortly thereafter sold to Dun & Bradstreet and discusses the role of ADAPSO in attempting to define the software industry and address legal and regulatory issues impacting the industry.

[Editor's Notes: This interview was conducted at an ADAPSO conference in Houston, TX. The official spelling of turnkey systems, inc. did not include capital letters. I have included them in this transcript to clearly identify that the reference is to a corporate entity and not to the term turnkey systems as a method of delivery for a combination of software and hardware products.]

Founding of Turnkey Systems

Luanne Johnson: Let's start off by talking about what motivated you to start a software company. When I went through the ICP Directories, Turnkey Systems showed up in April of 1969.

Lee Keet: Which has to be pretty early for a software product.

Johnson: Very early for a software product company. In the earlier Directories, in 1967 and 1968, there were banks listing software packages. There were service bureaus listing packages. There weren't very many companies that I would define as software products companies. Atlantic Software and Turnkey Systems were maybe two of the first that showed up. So what led up to your starting Turnkey Systems and getting into the software products business?

Keet: Well, I'll go back to IBM. Because I think the beginning of the software industry was IBM Type III programs. IBM had a program exchange called Type III programs which were programs contributed by customers or IBMers and made available to other IBM customers. Many of these were contributed by IBM systems engineers who were working in the field in branch and regional offices.

I was recruited to IBM right out of graduate school. I joined them in 1962 as a systems engineer but was immediately reassigned to something that was run by the Eastern Region of the United States and had its counterpart in the Midwestern and, I believe, the Western Regions which was called Advanced Application Development. It was the predecessor of IBM's Industry Marketing organization. Through the regional offices, IBM had gotten the idea that software was important. So they took people who had some software background and immediately assigned them to these application development activities. So I was in the White Plains offices of IBM very briefly before I was reassigned.

The work that I had done in school had led me into operations research and so I was looked at as the operations research maven and was assigned to a project which was intended ultimately to produce commercial software for manufacturing companies. When I say commercial, I do not mean that IBM intended to charge for it but that it was to be a package that was reproducible from customer to customer which would sell hardware. And in actual fact, we produced such a product. When I say "we", I don't mean my group exclusively. Several regions got together with the Midwestern Region as the lead and produced a product that became a bill-of-materials processor.

The thread of the bill-of-materials processor as a commercial product is a very interesting one to follow because it is the single leading applications product in existence today. Its offspring, MAPICS, etc., are all BMP based. You'll find references to the bill-of-materials processor, to DBMP as an extended access method, to a product called LAMP which was actually the commercial implementation of the materials planning package for the IBM 1440. Probably half of the 1440s were sold as a result of the LAMP product.

I first got involved in the development of the bill-of-materials processor in 1963 for the 1440. It was ported into the 360 almost immediately after its announcement in 1964 and I was

peripherally involved in that. The IBM Industry Marketing organization had been formed and I was moved into the manufacturing group run by a gentleman named Hal Schmidt who unfortunately died in an accident several years later. Our goal was to put together packages that would sell hardware. But the result of it was we actually put together software packages. And the only thing that distinguishes them from software packages today was that they were free.

Now, this was a bit of a skunk works activity within IBM because at the time all official software came through plants. In fact, that's still largely true for systems software today. It's plant-originated. This started as a regional effort and then was consolidated into Industry Marketing which was headquartered in White Plains. I stayed with them until I decided that I wanted some field sales experience. But I was looked at as a heavy-hitting engineer and not a guy who could ever sell anything. So I had to fight my way into a sales territory which ended up having nothing whatsoever to do with my prior background. It was a new account sales territory in Bridgeport which excluded anybody who had a computer or whoever wanted a computer. So I was out there selling unit record equipment and 632s and strange equipment to guys who were in businesses I had never heard of until I proved that I could sell. Then they gave me back the manufacturing territory.

But I retained the bug for software and when I got back to my manufacturing territory, I immediately went back to my friends in Industry Marketing and the region who knew all about LAMP, BMP and DBMP and made it my expertise to go sell software products, if you will, with hardware attached, to the large manufacturing accounts. And that's how I got really interested in software as an entity.

The 360 was such a success from a hardware sales standpoint and such a disaster from an installation standpoint that another IBMer and I decided that we needed to go into the software business to help customers get these machines installed. Our original goal was not software packages. Our original goal was custom contract development. In those days, we did not perceive a market for software packages. Nobody perceived a large market for selling something repetitively.

We started Turnkey Systems in April of 1967 and we were reasonably successful doing large custom development work and were one of the first developers of online systems in the Northeast. The 360 and the 2260 – the 2260 was IBM's first display terminal – had an obvious application for online order entry and online activity. Transaction processing was not really an accepted activity except in airline reservation systems. If somebody said, “What's an online system?” you always answered PARS, which was the airline reservation system. We got into some very interesting contracts developing online systems for local businesses in the Northeast and discovered that the software that was in between the application program and the hardware was non-existent.

There were some Type III programs, one of which was called CICS. And we looked at it and thought, boy, that's a piece of junk. And it was. It was developed by Commonwealth Edison in Chicago and then contributed to the IBM Type III program library. And there were a few others. But there was no official software that served that purpose. So we said, “Why don't we develop our own software package to interface IBM systems to 2260 environments?” We developed a program in 1969 which we used in one of our projects. It was so successful the customer said, “Why can't I use that for other application development?” And we said, “Sure, you can use it. You've already paid us a fee for it. Here. Take it.”

But it occurred to me at the time that we should set up a software products activity and see if we could sell this to other companies. So in 1969 I went on the road and tried to peddle this product which we called Graphics. Graphics was nothing more than an OS-only 2260 support package that allowed you to write Cobol programs and call Graphics which would in turn handle all the communications with the display. Basically what it was was a way that you could take an ordinary Cobol programmer and allow him to do something with a terminal which was very difficult for him. It had to all be done in assembler language. And you had to know how to format things.

Anyway, Graphics was a disaster. We sold about four of them and in the process discovered that if you really wanted to make a software package, it had to be OS and DOS interchangeable. It had to have multi-tasking capability. It couldn't stand there and take up the whole system while you waited. So we invented something that subsequently became known as pseudoconversationality which is not really multi-tasking or multi-threading. Well, it's multi-threading in the sense that when the program went away to ask the system to do something, we could go back to the operating system and act as if we were in a suspended state, as if our task had been completed. It's a very clever piece of technology but very simple.

So in January of 1970 we launched a product called Task/Master which was our first and probably most successful software package. It was to my knowledge the first commercial telecommunications monitor that was launched on a full worldwide basis. We went out and immediately signed up distributors in the Netherlands, Sweden, England, France.

Johnson: Was this in the 1970s?

Keet: In the very early 1970s. In late 1970, I think, we signed up Hoskins Systems in the U.K.

Johnson: Were you the first software product company to sign up international distributors?

Keet: No, I don't believe so. You might talk to Marty Goetz. I think ADR was ahead of us. We were awfully early, however. We had the pick of the crop in European distributors and we ended up with some good ones. We were getting a third of our sales in software products from outside of the United States in 1974 to give you a flavor for how fast we went into international markets.

Task/Master is still an active product. Some of the biggest banks in the world are still using it because it's fast. It's efficient. You can write Cobol very simply. And it's very hard to take out once you've got hundreds of applications. But we stopped selling it in 1978 which is another thing we did which I thought was extremely intelligent.

We recognized IBM as a force in this industry. This was a direct result of IBM having adopted CICS as its standard and slowly fixing it over the years to the point where it also became efficient. It also had a Cobol interface and so on. In 1978 we had a closed door meeting with our board and decided that we were going to exit that marketplace as graceful as we could without hurting our customers. We made an announcement to the customers that we would service them forever if they stayed on maintenance, that we would keep the system current to operating system standards. But it became essentially a defunct product. People still buy it if they have sister companies that are using it but we are not actively marketing it any more.

Challenges of Meeting Commitment to Customers

So, back to why we got into it. In 1969, there were four of us who went off to set up the software products activity. And we got hooked. There's no other way to put it. We saw that what we were selling to Company A led to enhancements that could be used for Company A and induce a sale to Company B. I guess if I went back I would say we couldn't get out of it because we were always running to meet our commitments. We were paying our bills by sales where we couldn't quite deliver what the next sale wanted. And we had to go back “into the lab” and add a feature. We did some – I won't say unethical things. But we did some things that betrayed the optimism of youth.

I remember selling Task/Master, one of the early ones, to American Tobacco. And we had announced it as a multi-tasking system. In actual fact, pseudo-re-entrancy is fine. Actual multi-tasking within a telecommunications monitor is frosting on the cake unless you're trying to run 20 transactions a second through the thing. American Tobacco didn't want to run two transactions a second through and they didn't need this feature. It was vaporware. It was announced but didn't exist.

So one day I got a call from the manager and he said, “We did a test and it's not truly multi-tasking.” I said, Oh, did you get the memo on that?” And he said, “No, I didn't get a memo.” So I said, “Well, I'll make sure you get a copy.”

We ran off and wrote Technical Memo Number 189 although, of course, there were no others in the series. It said that multi-tasking in this version had been disabled because of a serious design flaw and that it would be repaired in the next release of the product. I assembled my entire crew and we went back up to Oxford, Connecticut, where we rented computer time from Uniroyal and we invented the first multi-tasking system in the history of this business and shipped it to American Tobacco among others.

And that's the way you get hooked on this commitment to vaporware and fighting to keep up. I'm very pleased to say I don't think we ever disappointed a customer. And we never announced something we didn't deliver. But those were hairy days.

Johnson: There were some interesting gaps between announcement and delivery and some interesting things that occurred during those gaps.

Keet: Right. To end the story, in 1976 the software products business that I was running was paying all the bills of the company and had about $25,000 a month in profit. The other half of the business, which was the custom development business that my partner was running, was losing about $22,000 a month. And so we were marginally profitable on balance.

Our bank was attempting to eat our lunch and our dinner and so on. So we had a war meeting and I said, “Either you buy me out or I buy you out.” And the net of it was I bought him out for a lot of deferred payments and debt and so on which put the company into such an awkward position that it was obvious that our goal to be a company the size of Cullinet was going to be hampered by financing. So in 1978 we made the decision to find a partner and I sold the business to National CSS which shortly thereafter was sold to Dun & Bradstreet. I believed that size was going to be critical in the software business at that time and Dun & Bradstreet gave me the opportunity to go out and do things like buy McCormack & Dodge and do other things that built a reasonably sized nucleus. But that's in a nutshell the history of Turnkey Systems.

Defining a Software Product

Johnson: One of the things that I find interesting is that, in the 1960s, people were getting into the business of selling software products from some other kind of business. Nobody really knew what a software product was. As a matter of fact, I discovered in the meeting that was held [at the ADAPSO conference] yesterday that one of the challenges that still faces this

industry is to come up with a definition of what a software product is. Which I think is fascinating.

But at the beginning of the industry, everybody kind of made up their own mind as to what that product consisted of. Clearly a software product is not just a bunch of code. It's also all the support and training and documentation and whatever else you deliver with it. And then the question is: what kind of price do you put on it?

So tell me about the process of figuring out exactly what your product consisted of and how you figured out what it was worth. How did you figure out what you were going to do in terms of enhancements, ongoing maintenance and so on?

Keet: Well, I think we did it right. I think the answer would be if you went and found a sick business with respect to marketing you'd probably find out they weren't talking to the customers. That's how we decided what our product was. We built it. You might say we prototyped it. And then we took it out and tried to sell it and got rejected. We said, "Why did we get rejected?" And if it was anything other than we don't know your company or you're too small to do business with, if the answer was we don't like your documentation or you don't have a good PL1 interface or what have you, we would go back and decide whether we should build that and then incorporate that in the list of features of the product. When I say that, it sounds like a very formal process which in many companies it is.

But in the early days it was a daily process. Because the people that were developing the product were the same people who were designing the documentation and writing it. They were the same people who were designing the advertising campaigns and the marketing brochures. The same people who were getting on the phones and saying, "Can I come see you and tell you about this wonderful product?" And they were the same people who were going out and selling it and installing it.

We went from that nucleus of four to perhaps 15 within a year. But that group of 15 had very little defined job functions. Joe Farrelly, who's now Vice President of Research and Development at ADR, was Vice President of Research and Development at Turnkey Systems. In fact, the alumni of Turnkey Systems are all over the industry, another fact for which I personally am very proud. These are good people and they've really done very well. You can hardly find a major company that doesn't have one of the alumni.

There's a funny story about Joe. I sent him off to Atlantic National Bank in Florida with a new release to meet their needs. He would install it during the day and develop it during the night. And he would do that with a telephone line to our development people, meaning me and my guys in Oxford. We would develop along side of him and rush him code, sometimes reading it

to him over the phone, so that he could insert it and make things work the way they should. He would say we need this or the customer needs that and we would design it and get the code to him. Joe spent – he'll probably tell you it was five months, I think it was five weeks. I'm not sure the man slept.

Pricing Issues

The other part of the Atlantic National Bank story relates to your question about pricing. This was in the very early days. It might have been late 1970 but it was no later than early 1971. Atlantic National Bank said, "How much does your product cost?" We said, "$15,000." They said, "That's a lot of money. Do you lease it?" We said, "Sure." We ran back to the office to figure out what we could charge these guys on a lease basis. I did some quick calculations and said, "Let's offer them a three year lease for $500 a month and we'll be in good shape." So they signed a three year lease for the product for $500 per month, $18,000 over three years.

End of three years came. Atlantic National Bank said, "Now, what do we do?" So we said, "Well, do you want to buy it or sign another lease?" At this point we'd raised the price of the product to $25,000. And they said, "Wait. It was $15,000. Can't we buy it for $15,000?" And we said, "Sorry ,Charlie, it's $25,000". They said, "Well, what's your lease price now?" And we said, "Our lease price is now $1,000 a month." They said, "That's double what we've been paying." "Sorry, that's your choice. Or you can give it back." Which of course they couldn't do. So they signed another lease at $1,000 a month.

They finally bought the system about two years ago. They had been using it for 14 or 15 years. Somebody at the Turnkey Systems successor company figured out that they had paid $270,000 for that. I'm sure they got $270,000 worth of value but they could have bought it originally for $15,000. And maybe with maintenance and everything else added in, they would have paid $50,000 over that period of time. And they were a *bank*. And I won't even comment further.

But pricing was like that. You'd go out and see what the traffic would bear and then you'd adjust. Well, the way you adjusted was never to cut price. So you'd come in low. And if you had to cut price, you'd cut price – at least we would – by official discounts published on the price sheets. So a new user might get half off the lease rate for the first year and a right of cancellation for cause. That type of thing.

And those are still being used. In fact, ADR, now that they have this new found money from Ameritech, have just instituted exactly the same program which is the six month reduced rate "trial." And then we'd keep raising the price until we met resistance. At the end in 1978 when

we finally stopped aggressively selling it as a product, we were charging $78,000 for a product we brought out originally for $15,000.

Johnson: How'd you come up with the $15,000 in the first place? How did you determine that that's what the price should be?

Keet: It was sort of a random shot at what people would pay. We had this experience with Graphics where we had brought it to market at $7,500 and couldn't sell any of them. And we thought one of the reasons we couldn't sell it was the price was too low.

Johnson: Too low?

Keet: Too low. You should talk to John McGuire [of Software AG] on this subject. I remember an early ADAPSO meeting in New York City where John stood up and announced that he was going to sell a software product for $120,000 and not discount a penny off of it and everybody in the room laughed except me. I walked over to him later and said, “John, I don't have the balls to do that. But I'm going to watch you very carefully because I think you're on to something.”

Learning to Sell Software Products

I've always believed that a higher price attracts business. We had had an experience selling someone else's product in the late 1960s. It was called GRS and it was developed by Information Sciences which was a personnel consulting firm in its origins and is now a human resource package vendor.

Johnson: Is that InSci?

Keet: InSci, yes. They've had their ups and downs. But they were a very early developer of products and they developed a report writer called GRS.

Johnson: Was Dale Learn the founder?

Keet: Yes, Dale was the founder. And one of the people following Dale was a gentleman named John McGuffy. He called me up and said, “Would you like to market GRS for us?” This was before we had started the software products operation. And I said, “Well, we're thinking of selling software products.”

He said, “We’re selling GRS, but I'm trying to figure out how to distribute it.” I said, “I'll be your distributor for Connecticut just so we can learn.” We sold one to one of the insurance companies. I think it was the Hartford Insurance Company. And that's where I discovered that selling it is the easy part. John had said sell it for $15,000 – maybe that's where we got our number – and I'll give you half. And I said, “Gee, that's great.”

So I learned enough about it to demo it. Incidentally, it subsequently became a product called the Data Analyzer which I bought when I bought the company Program Products when I joined D&B. So it's still a very live and very active and very vibrant product. But GRS is long gone.

I sold it to The Hartford but I couldn't get the thing installed. I knew enough about how to sell it but I didn't know anything about installation, support, maintenance, training or any of the other things.

Johnson: This was before you were selling Task/Master or Graphics?

Keet: Absolutely. It was about the same time that we were trying to package up Graphics in 1969. So I called John and I said, “John, I just can't do this. I could sell it, but I don't want the customer to be disappointed. We're going to have to invest a lot of time to know about this product. And you've got some problems. There are bugs. The documentation's incomplete.” And I said, “I'm not moaning. But I want to give it back to you.” So, I gave him that sale and his people went up and installed it and did the training. The customer was very happy.

About three months later, John sent me $7,500. I called him up and said, “John, I didn't expect this.” He said, “I think you probably spent that much on your education. I got a check from the customer and I've learned a lesson that I'm going direct. I'm not going through agents in the U.S. So you keep the check. We'll both be happy.” I needed the money at the time so I thanked him profusely and immediately ran out and paid some creditors.

But that's how we learned a little bit about software products. I think that one of the reasons that the products that we developed were successful was that we gave them the trappings of a package. They came in a box. Originally the documentation was xeroxed stuff. But it was good xeroxed stuff. It was on paper with colored borders and was in a binder. It resembled what a lot of the early micro guys were putting out who were doing a good job.

Johnson: Did that $15,000 include installation? Did that include training?

Keet: That included my sister if you wanted her.

Johnson: (*Laughs*) In other words, are you telling me that you varied what was included depending on the customer's needs?

Keet: I'm trying to say that we would throw in what kinds of support it took to get the customer to buy. Frequently the customer would ask how much training was included and we'd answer two days. If he didn't ask the question we didn't offer anything. Ultimately our price sheets said two days of training were included and we would sell more if the customer wanted it. But in the early days we didn't know what we could sell the customer.

We made a terrible mistake and the industry is still making that mistake which is that the real business is product plus services. We didn't do that. We killed off our services business, closed it down. Even as late as when I bought it out and folded it up, I still did not understand that there was a huge follow-on support business. To my credit I'm not sure that there was as much of a business then because the customer didn't perceive the need. But today it's absolutely clear that the right place to be in the software product business is to almost have a product business as a loss leader to the professional services that you sell. Because these products have gotten so big and complex that you can buy a quarter of a million dollar general ledger package from MSA and the money is not in the quarter of a million dollar package which gets discounted heavily because of competition. It's in the half million to million dollars of follow-up services that that customer will buy from you in support, education, training, maintenance, etcetera.

Johnson: I guess what I'm trying to pin you down to is whether you had a specific list of items that were included in the price or whether your tailored that to each customer.

Keet: There was a price sheet. And the price sheet said you got one year of free maintenance, very much as it is today. If you buy it you get one year free maintenance. You get documentation. You get access to our technical people via a hotline. And you get a warranty which was repair of bugs. That's what you got for it. So the customer had a pretty clear idea of what he was buying

The product was defined both by the sales literature and also by a general information manual which was a real thin document but it described the capabilities of the product. It was more than a sales brochure but less than a user manual and it was referenced in our contract. So the customer had a pretty clear idea that it was a product. If you're trying to get to was it a product or a service, the customer thought he was buying a product. The services were only those that you felt were essential to use the product profitably.

We gave him a warranty which was pretty strong for those days. I think it still would be. And because we were little unknown company we gave him all sorts of other guarantees on financial disasters, or escrow software if he wanted. That's what I mean when I say my sister if you want. Anything that he wanted if it was logical in the transaction we would actually give him. We didn't necessarily volunteer everything because we weren't even sure what a software package should contain. The essentials were listed.

Competitors

Johnson: Who was your competition? Was it just IBM? Or did you have other competitors?

Keet: Oh, no. IBM was never a competitor. In the early days the competition was products that I'm not sure I can remember all their names.

Johnson: But there were other companies out there?

Keet: Oh, sure, that came in later. There was product that deviled us in the mid-1970s called Shadow which was a product from Altergo. It was a U.K. company that's subsequently been acquired by someone else. It went out of business because they had financial difficulties, to a large measure because they pursued the telecommunication commodity market too long. There was Datacom DB and Datacom DC which were the products that ADR bought from a company down in Texas called Insyte. Datacom DC was a separate telecommunications monitor that competed with us also in the mid-1970s environment. There was Model 203, I think, which was the data communications companion to Model 204, the data base system from Computer Corporation of America. Later in the cycle there was COM-PLETE which is the telecommunications monitor still being sold by Software AG. John Cullinane [of Cullinet] brought out a product called IDMSDC in approximately 1975, 1976.

Johnson: But you're still saying mid-70s. When did you first bring Task/Master out – 1969?

Keet: Well, the Graphics package was in 1969, Task/Master in 1970.

Johnson: So it sounds like you had a period of time there in which you didn't really have any serious competition.

Keet: We didn't have any major competition other than do it yourself or take the free access methods that IBM was distributing or the Type III programs that IBM was distributing and

use those. They were competitive in the sense that you got them through IBM. But they weren't actively sold as software packages.

Impact of IBM Unbundling

Johnson: What I'm trying to lead up to here is this belief in the industry that when IBM unbundled it created the industry. And whether that was because IBM was a serious competitor out there which Marty Goetz believes they were. Or whether it was because IBM conditioned the customer not to expect to pay for software from an independent vendor, whether it was more the psychological ramifications of the unbundling as opposed to the economic ramifications.

Keet: I would have to share some of the conventional wisdom which is that IBM created the market in the sense that it conditioned the customer to pay.

Johnson: You've gone on record as saying that IBM created the industry. Page three of your book. [*Preventing Piracy: A Business Guide to Software Protection*, Addison-Wesley, 1985.]

Keet: I believe IBM created the market. That doesn't mean that there wouldn't have been something out there that wouldn't have been good for everybody. But to give you our thinking at the time, I'll just tell you an anecdote.

When IBM announced unbundling it announced more than software packaging, pricing. It also announced that field SE services would be charged on an hourly basis. My immediate reaction to the unbundling announcement was not to go goody, goody, goody, we have a software industry. My immediate reaction was to run out and buy a drink for a guy who I really respected who was a field manager of SE Services for IBM and said, “Do you want to go into business with us?” So I was still thinking of it as a professional services business. And I thought unbundling created a new professional services industry. I didn't think unbundling created a software industry. And I don't think many people did at the time. It's only with hindsight.

Johnson: Let me show you something, for example. The myth persists. (*Refers to the document ADAPSO Research, Vol. 1, Issue 1, March 1985*, *Milestones in the History of the Software and Services Industry.)*

In here it says the surging industry revenue in 1969-70 was in large part due to IBM's unbundling of software. And then you look at the chart and you notice this incredible "surge"

(*chart shows modest growth in 1969 and 1970*). So the myth is there. The myth is there that the industry suddenly sprang to life at that point.

Keet: I think created is the wrong word. Legitimatized is the right word. What IBM did is it gave the customer the message that buying software is not only okay, it's required. And it also put a flag in front of the customer saying if you're going to spend your money you're going to have to justify the expense. And this didn't happen over night. That's the other thing.

The June of 1969 announcement was meaningful. But it was meaningful over a ten-year period. The customer had to start justifying an expenditure on IBM software as well as on ours. Ours had always been a one-off transaction. We would go all the way to the controller and sometimes to the president to get a sale simply because it was so unusual. What is it that you're buying? Well, we're buying a software package. Well, what is that? Why can't you do it yourself? And we had to justify it on the basis of productivity and so on. When IBM did it, the controller didn't want to see every transaction at that point.

So what happened was the buying authority got pressed down. The DP manager always knew what he needed. If he was smart enough he'd buy our product. But he didn't always have the authority. So the 1969 unbundling moved the purchasing authority for a $15,000 or $25,000 product down in the organization. And in fact, one of the price constraints that we always were sensitive to is where's the limit at which, in the average corporation, you have to go to the board of directors or to the president for purchasing?

So to a large measure the guys that broke that barrier, the DBMS guys, the John McGuires and John Cullinanes, were the guys that were willing to take their concept to a board and could explain it to a board. We never felt we could. How do you tell them what a teleprocessing monitor is? I've tried to do that for 20 years almost. And I still can’t. I mean, I can tell them all the good things about it. But I can't tell them what it is. My mother still asks me what I do. I send her glossy brochures and she puts them on her coffee table.

Johnson: I'm sure she's very proud of you. She just has difficulty explaining it.

Keet: She's absolutely in the dark as to why anybody would pay me to do anything and ever has.

Johnson: My father was the same way. He was very proud of the fact that I had started a software company but couldn't figure out what it was I was selling. Then my sister became a motel manager in Florida. He really thrived on that because he could get up early in the

morning and go out and count the cars in the parking lot. He could understand what it was that made the business go.

Keet: That's the way my mother is. We're taking her to Europe for two weeks in a couple of weeks. She's going on a TWA flight with us and she's flying first class. And she said to me, “This is terrific. You've got all this money to spend on me. But I don't understand why anybody would have given it to you. What is it that you do?”

ADAPSO’s Role in Software Issues

Johnson: I've been told that you were one of the first people to get involved with the issue of the accounting treatment of software.

Keet: The accounting treatment, the legal treatment, the taxation issue, I have been so frustrated by this. I walk away from a conversation and I come back to the conversation three years later and it's still the same sentence. Like Yogi Berra says, I feel like this is deja vue all over again.

I think software is a very mundane thing. I don't see it as much different than lots of other commercial products. It's not identical to phonograph records or compact disks or movies or books. But it shares characteristics with all those. It even shares characteristics with things like art work which can be reproduced, etchings and the like.

And the issue of SIC codes which is the one which just triggered me again – John’s memo on do we want to fight having ourselves classified along with other magnetically recorded media? Well, my position has always been, why? Join the mainstream, guys. It's going to help you. You're going to get the benefit of eons of tax thinking and accounting thinking. And you're going to get the benefits of investment tax credits, if you do this right. You're going to get the benefits of export/import supports. You're going to get the benefits of legal protections, copyrights and the like. Stop saying it should be treated in a sui generis manner. I have been a voice crying in the wilderness on that.

I wrote a little paper defining what software is after a lot of research. On ADAPSO's behalf, I polled all of the interested players. I got back reams of paper. I wrote a position paper on the definition of software. I took it to the SIA [Software Industry Association, a section of ADAPSO] board of which I was a member. And the SIA board said this is too hot for us to handle. Let's drop it. I voted with the majority on that because of the comments. It looked like we were going to run into internecine war over the definitional issue. Because everybody wanted it defined their own way.

They wanted it to be a service in California so it wouldn't be taxed. They wanted it to be a product, a machine component, ala Marty Goetz, so it could be patented. They wanted it to be tangible personal property under Section 38 or whatever it was so you could get the investment tax credit. And yet they wanted it to be an intangible -- well, I could go on and on. The point is the industry's still mired in this swamp.

Johnson: What's going to change that?

Keet: Maturity. The more Dun & Bradstreets that buy companies like McCormack & Dodge and the more American Expresses that get into this industry and so on, they are just not going to let their managers do that. In fact, as the professional managers come in, they're going to want accounting treatment and tax treatment that makes sense to them. They're not going to want some guy who's lived his whole life in the arcane world of software telling them that what he's got here is tangible but it's intangible. It's a product but it's a service.

Speculations on the Future of the Software Industry

Johnson: Do you think that this one thing is what makes this industry unique? Can you think of another industry that's been in existence for 20 years and can't define for itself what it is it's selling?

Keet: Well, it's because it's not an industry. There's no such thing as the computer services industry. This isn't an industry. This is a collection of technologists who have successfully sold their technology. But it will go away. There will be no such thing as the computer services industry in 20 or 40 years.

Johnson: You're saying that what exists now will be absorbed?

Keet: It's sort of like saying there's an engineering industry. There isn't an engineering industry. It's even almost like saying there's a manufacturing industry which people do. But there isn't really. Manufacturing an airframe is entirely different than manufacturing a computer chip. And then that's entirely different than manufacturing an automobile. And the three of them share nothing in common. Computer services and software really are the heart of every organization's life blood. The fact that there are independent suppliers of those, subcontractors if you will, sub-suppliers, well, I suppose that there may be people who are still independent suppliers of those components in the future. But I suspect if there are they will be parts of companies that supply the whole thing if you want it. Maybe an IBM or Honeywell or what have you.

The reason I say that is just look at the automobile industry. In 1920 you had Harvey Firestone trying to figure out the standardization of rims for tires. And he was making rims and making tires and selling both. And you had Delco making batteries for cars and lights, electrical systems. And you had Fisher making bodies. And you had Remy who was making the taillight assemblies or whatever. And you had a separate clutch manufacturers. They were all commercial enterprises. They all made money and they sold to each other and they sold to the automobile assemblers. Well, you know the story. General Motors is Fisher Body. General Motors is Delco Remy and so on.

I think there will be technological consolidation of the end result because is the customer really buying a telecommunications monitor? Is the customer really buying a data base manager? No. The customer is buying a component of a system.

As an example, you have McCormack & Dodge and what was called Dun & Bradstreet Computing Services which is the old NCSS and that's the revenues that ADAPSO counts. Say $150 million. But Dun & Bradstreet does $3 billion of revenue and I don't think that Dun & Bradstreet gets ten cents that isn't from computer-based service. Their *Official Airline Guide* is all computerized. In fact, many people are buying it electronically now rather than getting the print version. But the database is what you pay for. Moody's Investor Services is a large on-line, bond trading, bond rating service. Credit services is a large database of credit reports on people in businesses. Everything D&B does is in the computer services business.

So my point is either count them or recognize that the business they're in which they classify as business to business services or information services is dependent on computer services or computing. And there are suppliers who they rely on to provide some of those components. But that's not a separate industry. And as more of that goes into vertical markets which is what's happening – everybody's moving to vertical markets – it even loses the flavor of being a separate computer service.

There are people who are now packaging their complete application for an industry. And they're providing everything to that industry including training installation. They're in computer-based training. Is that computing services?

I may be overstating it when I say we're not an industry. But I don't think we're any more an industry than engineering is an industry or manufacturing is an industry.

Johnson: Okay.

Keet: Industries need to be defined on very clear axes. And it's usually defined by the customer base as one of the axes. And since we can't define our customers we can't call ourselves an industry. I also think one of those axes would be at least some general description of the product. And we certainly don't have that.

At Vanguard [Atlantic, Ltd., an investment firm Keet started after the sale of Turnkey Systems], we tried to define industries on three axes, customers, functions and technology. So you have to say: what's the technology base of the industry? That's difficult to do in our case. What are the customers we serve? That's difficult to do in what we do. And what are the functions? Can't do that either. I mean, it covers every single activity that involves computers. And every activity now involves computers. So we do everything. We're American industry. We're worldwide industry. I suppose you could say that there's some grandmother in Vermont who's knitting sweaters and selling them to her neighbors who is not included in our definition. But she probably has a little PC to keep track of her billings.

Johnson: Could be. It depends on whether she's hand-knitting her products or has an automatic knitting machine. That would have a chip in it.

Keet: I get pretty dogmatic on some of these points. And the reason I do, I guess, is because I was a very early participant in ADAPSO and other trade associations. And I do feel like it's deja vu all over again every time we discuss these things. Your positions get hardened. And then when I went to some of the first sessions with the micro software guys and they were talking about all the problems that they have. And they're the same problems we've been dealing with for years.

You know, when you can work in a batch environment where everything you do is constrained within the environment, it's possible to do glorious engineering or what appears to be glorious engineering to the user. But you have to remember that even the earliest IBM systems that had software products sold for them were multi-user even if they were sequential batch. And we were very rapidly into the online, multi-tasking, multi-threading, re-entrant programming modes where the delivery of very sophisticated software products from the end user's standpoint involved investments that then and today would be hundreds of times what it costs to develop the same system for micro.

The micro guys are just running into that now with LANs and with file servers and communication servers. And it's really hitting the fan for them. Because they're running into the same technological problems and the same cost problems and the same incompatibility problems between systems. You know, you put a Cytech LAN, which is what IBM sells, on your PC network and you're totally incompatible with anybody else's LAN. And there are people now making bridge LAN to LAN products. If you're selling a database management system you have

to say, which LAN will my software run with and so on? All those problems are the same problems we had in the 1960s on the mainframe. But we had to solve them before we could deliver a product. The PC guys didn't have to solve them because the early PCs -- you turned it on, you put your floppy in and you played with your spreadsheet. A spreadsheet is a nice piece of software but if you look at it from a technological complexity standpoint, it's trivial compared to some of the software we were trying to develop.

I'm fascinated by the industry. But I'm not sure that it isn't deja vu all over again. You should talk to Herb Jacobson if you haven't already.

Johnson: I haven't.

Keet: Herb is the father of Mark IV and of IMS.

Johnson: Oh, really?

Keet: Herb was on the original Mark IV development team which as you're aware was probably the first software product.

Keet: Yes, I've talked to John Postley [of Informatics which sold Mark IV]. I don't know that he mentioned Herb Jacobson.

Keet: John wouldn't necessarily mention the technical guy. Herb never got doodly squat.

Johnson: Do you know where he is now?

Keet: Yes, he's President of a company called Technology Information Products. And they are in Wellesley or Marblehead or one of the Boston suburbs.

Johnson: You also mentioned, yesterday I think it was, the origins of the bill-of-materials processor.

Keet: Well, I was going to try to remember the name of the guy in the Midwestern region. Because you probably aren't aware that Cincom Systems probably owes its entire existence to the bill-of-materials processor.

Johnson: No, I'm not aware of that.

Keet: The technology that they brought out as their original Total data base -- I may be mistaken on this and you should confirm it – was basically Control Sequential Access Method, CSAM. That was the underlying access method in BMP and DBMP. Tom [Nies of Cincom] will tell you, I think, that TOTAL was a totally unique engineered product. My external eye looked at it and said, it sure looks like CSAM. And they were DBMP experts of the Cincinnati IBM office.

Johnson: Cincom was originally called Cincinnati Computers.

Keet: Yes, right. They worked closely with a guy who was in Milwaukee whose name I can't remember who was the originator of the bill-of-materials processor.

Johnson: He was an IBM guy?

Keet: Yes, they all were. So was Tom. Tom was an SE.

Johnson: When you say that Herb Jacobsen was father of Mark IV and IMS...

Keet: He worked for Rockwell when Rockwell was developing IMS. IBM's leading products – with the exception of SQL and DB2, the new products – were all Type III programs. IMS was developed at Rockwell. Herb did IMS at Rockwell and then went to the Mark IV group. He's an interesting guy. And he was certainly around in those days and did a lot he never got compensated for. He's not bitter about it because he's not a bitter guy. But he really was very deeply involved in that.

Decision to Withdraw Task/Master from the Market

Johnson: Can you talk about your dispute with IBM?

Keet: Well, I'm not sure, but OK. It's an interesting story for the record anyway. The reason we backed out of selling Task/Master was we saw what I think is a trend that you'll see over and over again. We may have been the first people to see it which was that IBM announced that DL1, its data base system, would support online access using the master scheduler from CICS which means that if you wanted to use DL1 you had to buy CICS. Now, I'm no legal whiz. But that's a direct incontestable violation of the tie-in sale prohibitions of the

Sherman Act. And it also destroyed, in my opinion, the market. Even if the customer didn't want to buy DL1 he saw the handwriting on the wall that IBM was staking its claim to CICS.

Now, IBM had about 40% or 50% of the market at that point with CICS and the rest of us guys were arrayed in varying order. We were number two. We had sold about $25 million worth of Task/Masters at that point.

So I called my old friend Ed Kane [of IBM] and asked for a meeting. It was a closed door meeting. Grant Leschin was there. Ed Kane was there. And Alec – I've forgotten his last name. There were four of us in the room and I presented my case. I can't obviously remember the conversation verbatim but the gist of it was: Okay, you've got us. You're right. What are you going to do?

And I said, "Well, it seems to me there are two options, maybe three. I could sue you. I could forget the whole thing and decide that you destroyed my market. Or I could ask you to pay me off. I'm going to elect the second option but only on one condition. You assure me that what you did was unintentional. And you tell me that IBM is going to be more sensitive to what it did here in the future for the entire industry."

Well, I have a huge amount of respect for those guys, especially Ed Kane. And Ed said, "We didn't do this intentionally. We're really sorry about what it's done to you. You should feel free to pursue it in court if you think you need to." And I said, "Hey, my career is not to be a lawyer or to make my money suing IBM. Others may think that's their career – there was a time when everybody was suing IBM. Fortunately, I have other products now which I can pursue. But I think this is awful."

Well, an unnamed individual from IBM immediately offered me a large development contract with IBM which I turned down. Because I also didn't want to be IBM's slave. But it was very interesting from three standpoints.

First of all, I do honestly believe at the time that it was an IBM strategy to use tie-in relationships to capture market. I no longer believe that.

Secondly, I think it illustrated the naivete of IBM at the time in relation to the independent software companies. Because IBM would not knowingly have destroyed our market. I mean, there are other ways to do it without creating a prima facie case of an antitrust violation. That's one of those automatic treble damages things. So IBM would never have knowingly done that. It was just a huge blunder by a very smart elephant.

And, lastly, it illustrates the personalities. I think they were just good guys and still are.

However, I will say this for the record. I think that IBM has recognized the value of those implicit tie-ins. And that's the biggest single threat that faces the entire software industry. Because IBM has got a voracious appetite for revenue growth. And it can force the revenues into its camp as it did with CICS. CICS is the model that everybody should look at. It was an inferior product. Everybody acknowledged it. It was a deficient product in the sense that it didn't have even the interfaces the customers needed. But they would take it solely because IBM had announced that it was their only standard. In competitive battles you'd go in and the customer would say, “I have no choice.”

My model company is Hewlett Packard. You get in with a product. You are very technically innovative. You develop a market. You price it high. You get a lot of profits from it. And then when the guys come in with mass volume, the bigger guns come in, you step aside and bring up another product.

There are a lot of people who got crushed who said, I'm going to stay in and fight. Altergo went out of business. There are only two independent telecommunications monitors on the market now that I'm aware of. There were at least 30 ten years ago. You've got COM-PLETE which is really an extension of NATURAL and ADABAS. They don't sell a lot of those either. And you've got IDMSDC. And I don't think anybody's buying that. Those are the last remaining ones and they’re still there because those companies are so financially strong. Even ADR dropped Datacom DC. So it's just an example of what happens in a market where IBM says I want to have it. It also is an example of the component getting hidden. People don't really think any longer about making a competitive decision on a telecommunications monitor any more than you think about going out and buying a transmission for your car from somebody else. I'm sure you could. But I don't know where you'd go.

Johnson: Or why.

Keet: Or why. The early software products were just so awful that competitive analysis was important because it was a question of which one was worse than the other, which vendor was least likely to go out of business in the next six days. And it was really one that was done more on personal reputation. People were the thing you were selling because the products were so fragile. Now the products aren't fragile. The products and the companies are robust. And if you're going to buy from an MSA or McCormack & Dodge or UCCEL you don't ask to see the President of the company to determine whether you want to do business with them.

Johnson: Whether you trust them.

Keet: We used to have a magic rule that anything that hadn't sold eight copies wasn't a product. Just because there were so many things out there that had sold one or two copies in the early days. Because every one of them was a development site. After about eight independent installations you could call it a product.

Johnson: Well, that really covers the questions that I had. Are there any other comments that you want to make?

Keet: I have in my possession presented to me by my employees a copy of the original version of Task/Master which was all written in Cobol, believe it or not, with the original notes that I put in there. I helped write it. And I treasure it. But I think that things like that are more than memorabilia. They should go into the archives at the Babbage Institute.

As you pointed out, nobody has identified the personalities that were the early pioneers. I think there's great historical value in looking at it from the entrepreneurial slant, the birth of an industry, if you want to call it an industry, an industry. It can be a valuable road map for people.

I went to a meeting with the micro guys. There was a very distinguished panel. Bill Gates was chairing it. Dan Fylstra was on it. I forget who the others were but at the time they were all luminaries. This was in the hey day. The question came up of communications through the micro, multi-tasking, multi-threading. And one of the panelists made the statement that this is a very difficult technical problem, that they were going to have to wrestle with this thing for probably hundreds of man years.

Somebody in the audience, not me, raised his hand and said, "You know, I come from a mainframe environment. And that problem, pseudo re-entrancy and multi-threading and multi-tasking was solved on the mainframe in the early 1970s, maybe even the late 1960s. And there are road maps." A guy on the panel responded, "We don't want to even look at that. If we put that garbage in our minds we will not come up with the kinds of great solutions and answers that the micro world is producing."

Johnson: Isn't that an ironic statement to come from somebody who is selling a product in an industry that was based on the phrase, "Let's not reinvent the wheel." If I've heard that statement once, I've heard it a thousand times to support the need for software packages.

Keet: But in actual fact everybody did and is reinventing the wheel. The manpower that's going into software development today that is redundant because of the lack of standards is a disgrace. The only thing that justifies this is our entrepreneurial society. That mergers and acquisition conference we just came from? Some of those guys are going to be weak wolf cubs

who are eaten by their mothers. And others are going to become strong wolf cubs that survive on their own. But most of them are going to be absorbed into the pack. And with this constant turnover and the money just goes back and forth, back and forth, gets redeployed. I mean, you give the money to an entrepreneur and he takes the money and reinvests it in another company or becomes a venture capitalist. That's the only thing that justifies all this inefficiency. Well, that's all I have to say.

Johnson: Thanks, Lee. It’s been fascinating.